GoFast® Floating Point Libraries

Description

GoFast® is a family of high-performance floating-point libraries designed for embedded applications. They are ANSI C compatible and are designed to directly replace a C compiler's runtime floating-point support (library or coprocessor). GoFast® boosts the performance of an application's math calculations or eliminates the need for hardware floating-point coprocessors, in order to reduce product manufacturing cost. GoFast® libraries are reentrant.

Features

- Fast
- Reentrant
- ROMable
- Designed for real-time embedded systems

Floating Point Technology

GoFast is based on US Software's "Architecture Independent Technology" (AIT) and proven floating point algorithms that the company developed for over a decade. The algorithms have been thoroughly tested using automated methods. GoFast was specifically optimized and integrated for enhanced performance on each supported processor.

"Link and Go" Compiler Solutions

GoFast libraries are specifically integrated for "link and go" operation with each compiler. See the list of target processors and compilers supported, below.

Source Library Solutions

The GoFast IEEE 754 Floating Point Libraries are composed of single precision (32- bit format) and double precision (64-bit format) floating point routines delivered with full source code. Each library has been optimized for speed and space. Users are provided with immediate access to an array of floating point operations including add, subtract, multiply, divide, square root, sine, cosine, tangent, arctangent, exponentiation, log, natural log, and ASCII and integer conversion routines to and from the floating point format. Since source code is delivered, a GoFast® library can be customized to fit specific user application requirements.

Conformance and Testing

The accuracy of each GoFast Floating Point Library is within one (least significant) bit for arithmetic functions and two bits for transcendental functions, in most cases. The IEEE 754 Floating Point Format defines special representations for underflow, overflow, and invalid operation. The GoFast library routines use these formats and adhere to the IEEE 754 error handling procedures in all applicable cases. Quality assurance and testing procedures have assured proper product operation. In addition, each delivery includes target specific test programs assuring confidence of product installation and operation.

Target Processor and Compiler Support

68HC16
    GF-68HC16 includes:
        Archimedes/HI-Cross "Link & Go"
        Intermetrics[2] "Link & Go"
        Introl "Link & Go"
        Motorola Assembly Library
	ColdFire
	    GF-CF CodeWarrior "Link & Go"

680x0 / 683xx
    GF-M68K includes:
        GNU "Link & Go"
        Intermetrics "Link & Go"
        Microtec "Link & Go"
	MIPS
	    GF-MIPS32 includes:
	        MIPS SDE / GNU "Link & Go"

8051
    GF-8051 Franklin/Keil Interface
	NEC V8xx
	    GF-V83x Green Hills "Link & Go"
	    GF-V85x Green Hills "Link & Go"

80x86 Real Mode
    GF-BCC Borland "Link & Go"
    GF-IC86 Intel iC86 "Link & Go"
    GF-MSC Microsoft "Link & Go"
    GF-MS72291 MSC to NEC 72291
	Nios II
	    GF-NIOSII-GNU GNU "Link & Go"

80286 / 16-bit Protected Mode[3]
    GF-286 includes:
        Borland "Link & Go"
        Microsoft "Link & Go"
	Power PC
	    GF-PPC includes:
	        Diab Data "Link & Go"
	        GNU "Link & Go"
	        IBM "Link & Go"
	        Metaware "Link & Go"

80386 / 32-bit Protected Mode[3]
    GF-HIGHC Metaware "Link & Go"
    GF-MPROT Microsoft "Link & Go"
    GF-PROT Non-compiler-specific library
    GF-WCC Watcom "Link & Go"
    GF-WR WindRiver GNU Assembly Library
	SH 1, 2
	    GF-SH7xxx includes:
	        GNU "Link & Go"
	
	SH 3
	    GF-SH3 includes:
	        GNU "Link & Go"
	        Hitachi C "Link & Go"

8096 / 80196
    GF-80196 BSO[1] C Interface
	SPARC / SPARCLite
	    GF-SPARC includes:
	        GNU "Link & Go"
	        Green Hills "Link & Go"
	        Microtec "Link & Go"
	        SUN "Link & Go"

ARM
    GF-ARM-IAR IAR EWARM "Link & Go"
	Z80 / Z180 / 64180
	    GF-Z80 IAR/Archimedes "Link & Go"

Notes:
1. BSO 8096 C compiler was formerly the Intel iC96 C compiler.
2. The 68HC16 Intermetrics C compiler is a Whitesmith C compiler.
3. x86 Protected Mode versions assume use of the Phar Lap DOS Extender.